Penerapan Algoritma Rijndael untuk Mengamankan Teks

Tiepsi [1], Kristian Siregar[2]

[1,2] STMIK Budi Darma, Jl. Sisingamangaraja No.338 Simpang Limun Medan

ARTICLE INFORMATION

Received: Agustus 23,2019
Revised: September 20,2019
Available online: Oktober 03,2019

KEYWORDS

Kriptografi, Algoritma Rijndael, Penyandian Teks.

CORRESPONDENCE

Phone: +6283160861655

E-mail: Tiepsi.dalimunthe@yahoo.co.id

A B S T R A K

Kriptografi merupakan salah satu solusi yang dapat dimanfaatkan dan dikembangkan dalam menghadapi permasalahan tentang keamanan data. Salah satu algoritma yang bisa dihandalkan dalam keamanan data adalah algoritma Rijndael. Algoritma Rijndael merupakan algoritma kriptografi simetrik yang beroperasi dalam mode penyandi blok (block cipher) yang memproses blok data 128-bit dengan panjang kunci 128 bit, 192 bit, atau 256 bit.Untuk menjamin keamanan dan keutuhan dari suatu data, dibutuhkan suatu proses penyandian, dengan melakukan proses enkripsi dan dekripsi. Proses enkripsi ini akan mengubah suatu data asal atau asli menjadi data rahasia yang tidak dapat dibaca. Sementara itu, proses dekripsi dilakukan oleh penerima data yang dikirim tersebut. Data rahasia yang diterima akan diubah kembali menjadi data asal atau asli. Dengan cara penyandian tadi, data asli tidak akan terbaca oleh pihak yang tidak berkepentingan, melainkan hanya oleh penerima yang memiliki kunci dekripsi.

1. PENDAHULUAN

Kriptografi merupakan studi matematika yang mempunyai hubungan dengan aspek keamanan informasi seperti integritas data, keaslian entitas dan keaslian data. Kriptografi menggunakan berbagai macam teknik dalam upaya untuk mengamankan gambar. Penyandian gambar melalui media elektronik memerlukan suatu proses yang dapat menjamin keamanan gambar dan keutuhan dari gambar tersebut. Gambar tersebut harus tetap rahasia dengan bertujuan untuk menjaga karahasiaannya terhadap akses orang-orang yang tidak berhak[1].

Masalah keamanan dan kerahasiaan pada suatu gambar merupakan hal yang sangat penting dalam suatu organisasi maupun pribadi terlebih apabila gambar tersebut berada dalam suatu jaringan komputer yang terkoneksi dengan jaringan publik misalnya internet. Tentu saja gambar yang sangat penting tersebut tidak sembarangan dilihat atau dibajak oleh orang yang tidak berwenang. Apabila hal ini sampai terjadi kemungkinan gambar akan rusak bahkan dapat hilang dan akan menimbulkan kerugian material yang besar[2], [3].

Algoritma Rijndael merupakan algoritma kriptografi simetri yang beroperasi dalam mode penyandi blok (block cipher) yang memproses blok data 128-bit dengan panjang kunci 128-bit (AES- 128), 192-bit (AES-192), atau 256-bit (AES-256). Oleh karena itu untuk melindungi kerahasiaan data yang tidak aman, penulis menggunakan metode algoritma Rijndael untuk proses enkripsi dan deskripsi data text. Kriptografi telah menjadi suatu bagian yang tidak dapat di pisahkan dari sistem keamanan data, Salah satu metode enkripsi data adalah algoritma Rijndael yang merupakan algoritma cipher blok yang popular karena dijadikan standar algoritma enkripsi kunci-simetri[4]..

Adapun beberapa manfaat dari penelitian ini adalah melindungi dan merahasiakan data yang akan dikirim dengan menggunakan algoritma Rijndael, Meningkatkan keamanan teks dari tindakan-tindakan pengerusakan atau hal-hal lain dari orang-orang yang tidak bertanggung jawab, Meningkatkan proses pengamanan teks melalui sebuah aplikasi.

2. LANDASAN TEORI

Kriptografi pada awal nya dijabarkan sebagai ilmu yang mempelajari ilmu bagaimana menyembunyikan pesan. Namun pada pengertian modern kriftografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi dan entitas[5]

Rijndael diciptakan oleh dua orang kriptografer asal Belgia, Vincent Rijmen dan Joan Daemen. Rijndael adalah suatu blok cipher yang terdiri dari sebuah variabel blok dan sebuah variabel kunci dengan panjang blok dan panjang kunci dapat secara sepesifik ditentukan 128, 192 atau 256 bit [4], [6].

a. Proses Dekripsi

Transformasi-transformasi yang merupakan kebalikan dari setiap cipher diterapkan dalam program dekripsi (inverse cipher). Fungsi AddRoundKey() untuk enkripsi digunakan kembali untuk dekripsi (Muhammad Farid Fachrurozi, 2006, 4) Adapun yang harus dibuat lagi adalah:

1. InvSubBytes(),

2. InvShiftRows(),
3. InvMixColumns().

Rentetan tersebut dijalankan sebanyak Nr-1 sebagai loop utama, setiap loop disebut round (Nr = 10 round untuk Rijndael-128). AddRoundKey() dieksekusi sebagai round ini sial sebelum loop utama. Setelah loop utama tersebut berakhir (sembilan round), SubBytes(), ShiftRows(), dan AddRoundKey(), dieksekusi secara berturut-turut sebagai final round.

b. Proses Dekripsi

Transformasi-transformasi yang merupakan kebalikan dari setiap cipher diterapkan dalam program dekripsi (inverse cipher). Fungsi AddRoundKey() untuk enkripsi digunakan kembali untuk dekripsi .Adapun yang harus dibuat lagi adalah:

1. InvSubBytes(),
2. InvShiftRows(),
3. InvMixColumns().

Beberapa bagian cukup dikopi dari fungsi kebalikannya yang telah digunakan saat enkripsi. AddRoundKey() dieksekusi sebagai initial round, diikuti sembilan round rentetan InvShiftRows(), InvSubBytes(), InvMixColumns(), dan AddRoundKey(). Round ke-10 yang mengikutinya tidak menyertakan InvMixColumns serupa dengan final round enkripsi[7].

3. HASIL DAN PEMBAHASAN

Analisis adalah penguraian dari suatu pembahasan, dalam hal ini pembahasan mengenai analisis algoritma rijndael pada proses pengamanan data teks yang berguna untuk mengetahui proses enkripsi dan dekripsi dari kedua algoritma rijndael. Permasalahan utama yang ada pada perangkat lunak adalah mengenai enkripsi dan dekripsi. Bagaimana melakukan enkripsi terhadap sebuah data teks pada proses pengamanan dekripsi terhadap data teks yang sudah berisi ciphertext sehingga dapat ditampilkan ke user dalam plaintext. Dimana data yang ada pada komputer boleh jadi merupakan data yang sangat penting dan rahasia yang tidak dimungkinkan orang lain mengaksesnya.

Kriptografi bertujuan untuk menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak berkepentingan. Kebanyakan kriptografi diterapkan pada file yang memiliki format.txt, .doc, dan .pdf. Masalah yang timbul adalah bagaimana jika data/pesan rahasia tersebut yang ada di komputer tidak mudah dapat dibaca oleh orang lain yang tidak berkepentingan sehingga dapat menjadi suatu hambatan apabila komputer kita sudah terhubung pada saat mendistribusikan file teks yang berisi data/pesan rahasia tersebut.

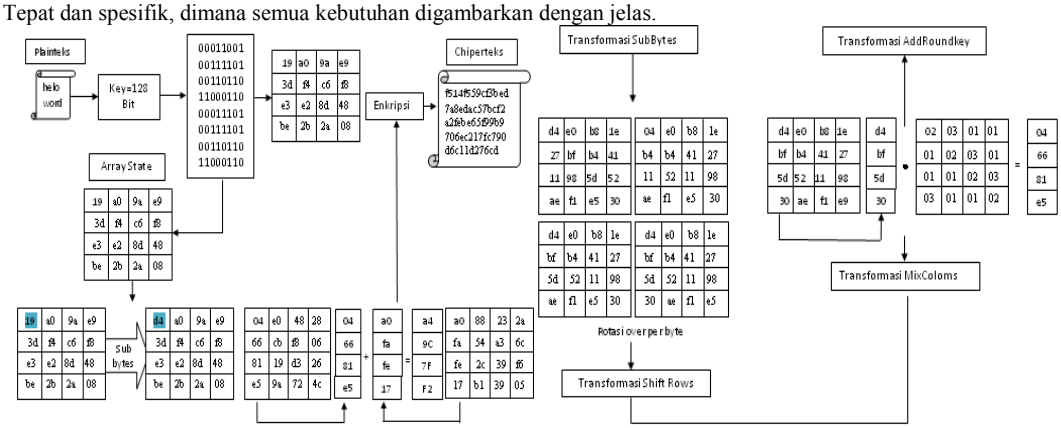
a. Proses Enkripsi Rijndael

Algoritma Rijndael juga melakukan putaran enkripsi (enciphering) sebanyak 10 putaran namun bukan putaran yang merupakan jaringan Feistel. Enciphering pada Rijndael melibatkan empat proses yaitu :

1. Sub Bytes
2. Shift Rows
3. Mix Columns
4. Add Round Key

Secara umum, proses enkripsi dilakukan dengan initial round yaitu melakukan XOR antara state awal yang masih berupa plaintext dengan cipher key. Kemudian melakukan keempat proses diatas sebanyak 9 kali putaran, dan terakhir adalah final round yang melibatkan proses sub bytes, shift rows, dan add round key. Tujuan analisa sistem ini adalah menentukan kebutuhan pemakai secara akurat yang berarti :

1. Semua informasi yang di inginkan telah dimasukan secara lengkap
2. Mudah dimengerti oleh pengguna
3. Konsisten
4. Tepat dan spesifik, dimana semua kebutuhan digambarkan dengan jelas.



Gambar 1 : Proses Enkripsi Rijndael

Langkah-langkah dari proses enkripsi Rijndael diatas adalah sebagai berikut:

Plainteks = Helo word
Kunci = 128 Bit

1. File teks yang akan dienkripsi dengan format teks yang digunakan adalah txt, doc, dan pdf. Sedangkan untuk ukuran file nya tidak ada batasannya, semakin besar ukuran file nya maka semakin banyak pula pesan yang dapat dienkripsi.
2. Pada proses enkripsi Rijndael proses Key Schedule sama, mengambil kunci cipher dan melakukan rutin ekspansi kunci (key expansion) untuk membentuk key schedule.
3. Merubah karakter plainteks kedalam bilangan biner. Maka hasil biner dapat dilihat pada tabel 1 di bawah ini.

Tabel 1. Plainteks dalam Biner

0	0	0	1	1	0	0	1
0	0	1	1	1	1	0	1
0	0	1	1	0	1	1	0
1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	1
0	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0

Hasil biner diatas akan dilakukan proses selanjutnya yang menggunakan Array State yang proses adalah sebagai berikut. Plainteks yang berukuran N ROWS NCOLS. Elemen Array state mengacu pada s[r,c]=in[r+4c] untuk $0 \leq r < 4$ dan $0 \leq c < Nb$. Maka hasil dari pada Array State dapat dilihat pada tabel 2 di bawah ini:.

Tabel 2. Array State dari Biner Plainteks

19	A0	9a	E9
3d	f4	C6	F8
E3	A2	8d	48
Be	2b	2a	08

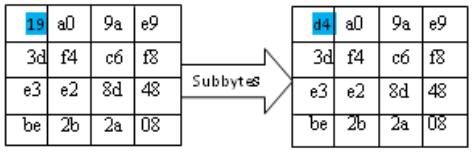
Dari Gambar 3.3 dapat dilihat bahwa (Nb adalah panjang blok dibagi 32). Pada AES, Nb = 128/32 = 4. Tiap elemen dari array state diisi dengan 8 bit teks (1 byte) dalam notasi HEX.

4. Setelah pengisian array state selesai, lakukan transformasi SubBytes dimana transformasi byte pada setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam Tabel 1

Tabel 3 S-Box.

Hex	Y															
	0	1	2	3	4	5	6	7	8	9	a	B	c	d	e	f
X 0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	Af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	Ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	Fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	f2
8	cd	0c	13	Fc	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	Dc	22	2a	90	88	46	ee	b8	14	de	5e	0e	Db
A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	Ea	65	7a	ae	08
C	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
D	0	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
E	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	Bf
F	8c	a1	89	0d	bf	bf	42	68	41	99	2d	0f	b0	54	bb	16

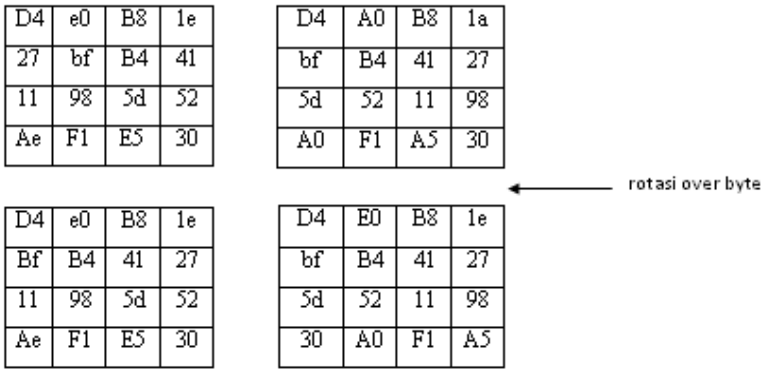
Berikut hasil array state yang sudah di transformasi SubBytes :



Gambar 2. Transformasi Sub Bytes

Dilihat bahwa S-box yang digunakan dalam SubBytes() transformasi disajikan dalam bentuk heksa desimal pada Tabel 1 di atas.

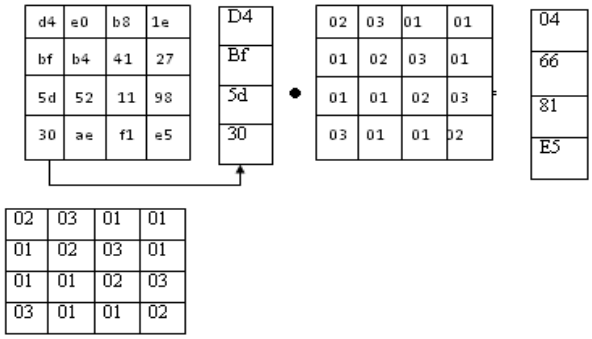
5. Lakukan transformasi Shiftrows yang pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bi). Proses pergeseran ShiftRows ditunjukkan pada Gambar 3 berikut :



Gambar 3 : Transformasi Shiftrows

6. Setelah proses transformasi ShiftRows selesai, lakukan pengacakan MixColumns dengan cara melakukan perkalian matriks yang merupakan transformasi dari perkalian polinom antara tiap kolom dengan polinom 4 suku pada GF(28),

$a(x) \bmod (x4+1)$.



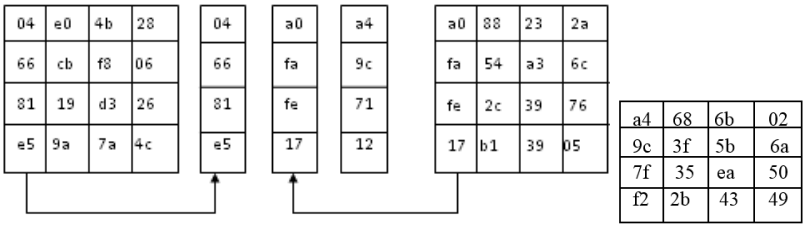
Gambar 4 : Pengacakan MixColumns

7. Pada proses dekripsi Rijndael proses AddRoundKey sama, sebuah round key ditambahkan pada state dengan operasi XOR. Transformasi AddRoundKey pada proses enkripsi pertama kali pada round = 0 untuk round selanjutnya round= round + 1

04	E0	48	28
66	Cb	F8	06
81	19	D3	26
E5	9a	7a	4c

A0	88	23	2a
fa	54	A3	6c
fe	2c	39	76
17	b1	39	05

Berikut hasil dari *array state* dan *round key* yang sudah di transformasi *AddRoundKey*:



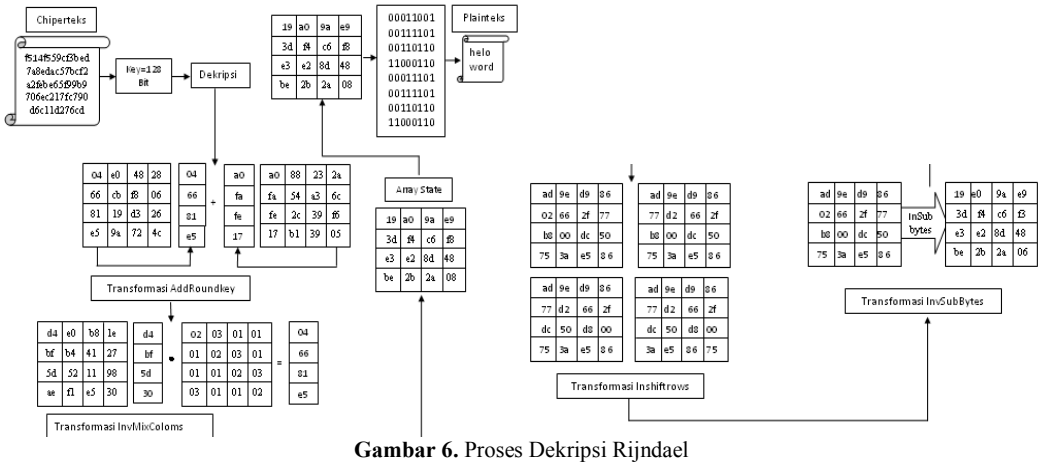
Gambar 5. Transformasi *AddRoundKey*

8. Setelah selesai melakukan semua langkah diatas maka simpan *file* tersebut sebagai *cipherteks* (*file* teks yang sudah terenkripsi).
9. Maka hasil dari proses enkripsi dapat dilihat dari berikut ini.

Tabel 3. Hasil Enkripsi

F5	14	F5	59	cf	3b	Ed
79	8e	da	C5	7b	cf	2
A2	fe	be	65	F9	9b	9
70	6e	C2	17	fc	79	0
D6	C1	1d	27	6c	df	

b. Proses Dekripsi Rijndael

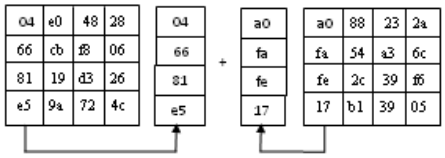


Gambar 6. Proses Dekripsi Rijndael

Langkah-langkah dari proses dekripsi *Rijndael* diatas adalah sebagai berikut:

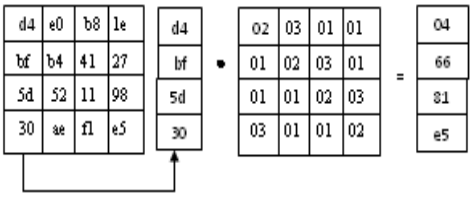
1. Proses pertama yang dilakukan adalah *file* teks yang telah dienkripsi dengan menggunakan algoritma *Rijndael* untuk di dekripsi agar mudah dibaca pesan yang tersimpan didalamnya.
   f514f559cf3bed
   7a8edac57bcf2
   a2febe65f99b9
   706ec217fc790
   d6c11d276cd
2. Pada proses dekripsi *Rijndael* proses nya sama, mengambil kunci *cipher* dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*.

   Key=128 Bit
3. Selanjutnya Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada *round* = 0 untuk *round* selanjutnya *round*= *round* + 1.



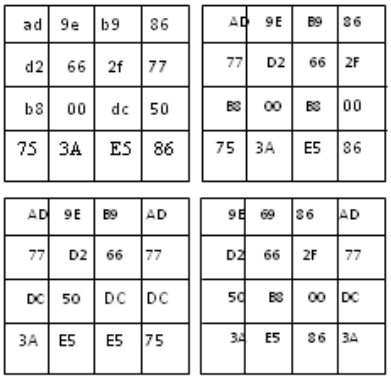
Gambar 7. Transformasi *AddRoundKey*

Setelah melakukan proses *AddRoundKey*, lakukan pada setiap kolom dalam *state* dikalikan dengan matriks *MixColumns* dalam perkalian *Rijndael*.



Gambar 8. *MixColumns*

4. Setelah melakukan perhitungan pada perkalian matriks di *InvMixColumns* dilanjutkan transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*, karena pada transformasi *InvShiftRows* dilakukan pergeseran *bit* ke kanan.

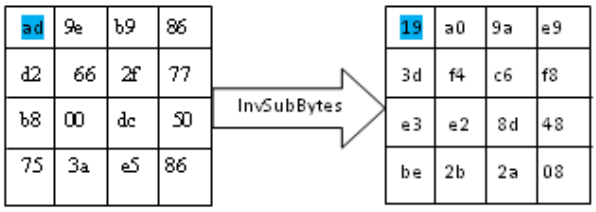


Gambar 9. Transformasi InvShiftRows

5. Dilanjutkan proses transformasi *InvSubBytes* dimana transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*, karena pada *InvSubBytes* tiap elemen pada *state* dipetakan dengan menggunakan tabel *Inverse S-Box* yang ditunjukan dalam.

Tabel 4. Tabel *Inverse S-Box*

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	Xe	xf
x0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	Fb
x1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	Cb
x2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
x3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6b	8b	d1	25
x4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
x5	6c	70	48	50	fd	Ed	b9	da	5e	15	46	57	a7	8d	9d	84
x6	90	d8	ab	00	8c	Bc	d3	0a	f7	e4	58	05	b8	b3	45	06
x7	d0	2c	1e	8f	ca	3e	0f	02	c1	Af	dd	03	01	13	8a	6b
x8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
x9	96	ac	74	22	e7	Ad	35	85	e2	f9	37	e8	1c	75	Df	6e
Xa	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	Be	1b
Xb	fc	56	3e	4b	c6	d2	79	20	9a	Db	c0	fe	78	cd	5a	f4
Xc	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	Ec	5f
Xd	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	Ef
Xe	a0	e0	3b	4d	ae	2a	f5	b0	c8	Eb	bb	3c	83	53	99	61
Xf	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d



6. Setelah transformasi *InvSubBytes* sudah dilakukan, maka hasil *array state* yang awal enkripsi akan muncul kembali untuk dijadikan status awal dan dikalkulasikan c*ipherteks* menjadi *plainteks* kembali.

Tabel 5. Hasil array state

19	a0	9a	e9
3d	f4	c6	f8
e3	a2	8d	48
be	2b	2a	08

7. Merubah karakter plainteks kedalam bilangan biner. Maka hasil biner dapat dilihat pada tabel 6 di bawah ini.

Tabel 6. Hasil Biner Array State

0	0	0	1	1	0	0	1
0	0	1	1	1	1	0	1
0	0	1	1	0	1	1	0
1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	1
0	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0

8. Setelah selesai melakukan semua langkah diatas maka simpan *file* tersebut sebagai *plainteks* (*file* teks asli).
9. Maka hasil dari proses dekripsi dapat dilihat pada tabel berikut ini.

Tabel 7. Hasil Dekripsi

h	E	l	0
w	O	r	D

4. KESIMPULAN

Setelah menyelesaikan perancangan perangkat lunak enkripsi dan dekripsi, penulis menarik kesimpulan sebagai berikut:

1. Proses pengamanan algoritma *Rijndael* dapat dilakukan dengan langkah-langkah antara lain *SubBytes*, *ShiftRows*, *MixColumns*, dan *addRoundkey*.
2. Penerapan Algoritma Rijndael dalam mengamankan Teks telah berhasil mengamankan Teks dari penelitian yang penulis lakukan.

DAFTAR PUSTAKA

[1] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.

[2] A. Suhendra, "Steganografi Pada Citra Terkompresi Metode Huffman," *MEANS (Media Inf. Anal. dan Sist.*, vol. 1, no. 2, pp. 33–39, Dec. 2016.

[3] B. Silaban and T. Limbong, "Aplikasi Pembelajaran Pengenalan Kriptografi Algoritma Affine Cipher Dan Vigenere Cipher Menggunakan Metode Computer Assisted Instruction," *Media Inf. Anal. dan Sist.*, vol. 2, no. 2, pp. 14–20, 2017.

[4] J. Daemen and V. Rijmen, *The Block Cipher Rijndael*, vol. 1820. 1998.

[5] A. M. Hasibuan, "Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AES Pada Smartphone," *MEANS (Media Inf. Anal. dan Sist.*, vol. 2, no. 1, pp. 29–35, Jun. 2017.

[6] D. D. Santoso and P. Tarigan, "Penerapan Algoritma Playfair Cipher sebagai Penyandian Kunci Dalam Pengamanan File Teks dengan Algoritma Rijndael," *Pelita Inform. Budi Darma*, vol. 17, pp. 59–64, 2018.

[7] M. Mankar, "Encryption and Decryption Using Rijndael Algorithm," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 4, pp. 1387–1391, 2015.